**Feature selection para el dataset Sonar**

Feature selection es una estrategia muy importante para generar modelos buenos y robustos. Si mantenemos atributos que contengan una gran cantidad de ruido, identificar los patrones que se buscan en un set de datos será una tarea difícil, hasta para los algoritmos de aprendizaje más modernos.

Por este motivo, existen diferentes técnicas para seleccionar los mejores atributos para nuestro modelo. Estas caen en 2 categorías, las del tipo filtro y las del tipo wrapper.

Las técnicas del tipo filtro refieren a una elección de artibutos a remover sin incluir el proceso de aprendizaje/modelado. Dentro de esta categoría, podemos encontrar a la eliminación de atributos por conocimiento del dataset o a la eliminación de atributos conocidos que no aportan a los modelos (id o nombre en algunos casos). También, técnicas como el análisis de la correlación de los atributos con la salida y entre sí, caen dentro de esta categoría. Por último, otra técnica de tipo filtro sería utilizar PCA, tanto para utilizar los primeros componentes principales en el modelado como para identificar los atributos con coeficientes mayores en el cálculo de los primeros componentes principales.

Por otra parte, las técnicas de tipo wrapper seleccionan predictores de forma iterativa incluyendo los pasos de entrenamiento y validación para decidir cuáles predictores son los que generan los mejores modelos. Como una estrategia de fuerza bruta, o sea probar todas las combinaciones posibles es increíblemente costoso de un punto de vista computacional (orden exponencial – 2 a la n para n atributos), surgen heurísticas variadas para atacar esta problemática. Algunos ejemplos de estas son: forward propagation, backward elimination y evolutionary feature selection. Estas heurísticas serán analizadas a continuación.

**Forward Selection**

Como primer paso, se generan modelos con cada uno de los predictores del dataset (se utiliza únicamente 1 predictor). El atributo con el que fue entrenado el modelo que obtuvo mejores resultados, será seleccionado y se pasa a la siguiente iteración. Luego, se entrenan modelos con 2 atributos, las combinaciones entre el atributo elegido en el paso anterior y el resto de los predictores (n-1 modelos con n predictores). Dentro de estos modelos, se compara la performance entre sí y con el modelo que utilizó únicamente 1 atributo. Si la performance no mejora respecto al modelo con 1 atributo menos, termina la heurística. Si, efectivamente, se logra obtener una mayor performance, se sigue iterando hasta llegar a la condición de corte mencionada en la oración anterior. Se puede ver que el número de entrenamientos necesarios para esta heurística posee una cota superior dada por la suma de los primeros n naturales de 1 hasta la cantidad de predictores. Esta cantidad de modelos entrenados es mucho menor que la cantidad si utilizamos el enfoque de fuerza bruta (para 100 atributos existen como máximo 5050 combinaciones para forward selection y 1,267,650,600,228,229,401,496,703,205,375 para el enfoque de fuerza bruta).

**Backward elimination**

Esta heurística se puede pensar como la opuesta/inversa de forward selection. En vez de empezar con 0 e ir agregando atributos iterativamente, aquí se comienza con la totalidad de los predictores. Luego de entrenar el primer modelo con todos los atributos, se mide la performance de los modelos que poseen todos los predictores menos 1. Si existe alguno de estos nuevos modelos que posea una mejor performance que la que obtuvo el modelo de la iteración anterior (modelo entrenado con todos los predictores), se descarta el atributo no utilizado y se itera nuevamente. En el caso que la performance no mejore, termina la heurística.

Las 2 heurísticas mencionadas son claramente mucho más eficientes desde el punto de vista computacional que la fuerza bruta, pero tienen sus desventajas. Son muy susceptibles a seleccionar óptimos locales y no el óptimo global. Debido a esto, la performance obtenida con estos enfoques normalmente es inferior a la obtenida con el enfoque de fuerza bruta. Entonces, ¿cuál sería nuestra opción si quisiéramos lo mejor de los dos mundos?

**Algoritmos Evolutivos para la selección de atributos.**

Los algoritmos evolutivos son una técnica de optimización que se basa en la selección natural. En resumen, esta estrategia funciona inicialmente generando una población inicial (combinaciones de predictores en nuestro caso). Luego, de forma iterativa hasta llegar a un máximo de iteraciones (se les llama generaciones) o a alguna otra condición de corte, se aplican los siguientes pasos. Cruzamiento: Se cruzan los individuos de la población. Mutación: se muta a los individuos (normalmente involucra un cambio pequeño randómico como seleccionar un atributo nuevo o des-seleccionar uno). Evaluación: se evalúa a la población. En nuestro caso esto se hará con la exactitud (por ejemplo) de los modelos generados con los atributos representados por cada individuo de la población. Selección: se seleccionan losindividuos que posean mejores métricas.

Esta estrategia, aunque tiene un costo computacional mayor que las 2 heurísticas antes mencionadas, suele llegar al óptimo global o de los contrario a una solución mejor que la obtenida con dichas heurísticas. El costo igualmente, no se compara con el de la fuerza bruta.

**Dataset Sonar**Gráfico, Gráfico de superficie

Descripción generada automáticamente

Este dataset contiene medidas realizadas al fondo marino utilizando un sonar. Cada medida (ejemplo) contiene múltiples bandas de frecuencia (predictores) y una clase definida: rock o mine. La idea es poder detectar la presencia de minas en el fondo marino. Al graficar los datos, se puede observar el espectro de frecuencia para cada objeto. Dichos espectros, difieren claramente en algunas zonas específicas. Se definen los siguientes intervalos de atributos como zonas de interés. La idea es utilizar diferentes técnicas de feature selection para observar los atributos que se eligen y la performance que se obtiene.

**Intervalos**

1. 9-13
2. 19-25
3. 33-37
4. 42-50

## Ejercicio 1: Benchmarking

Se entrena un modelo con todos los predictores y el algoritmo naive bayes.

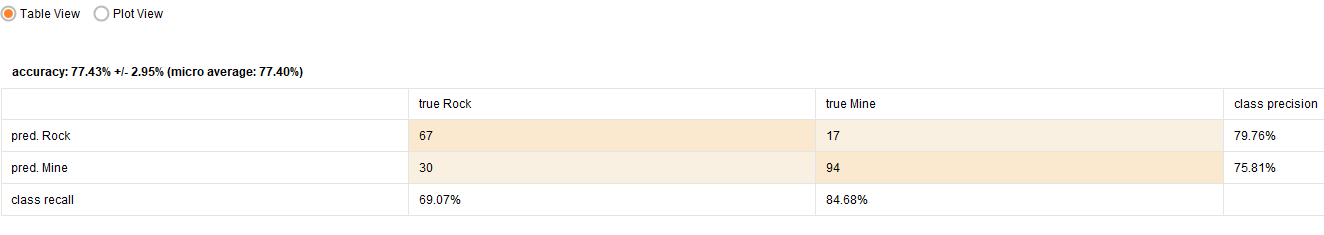
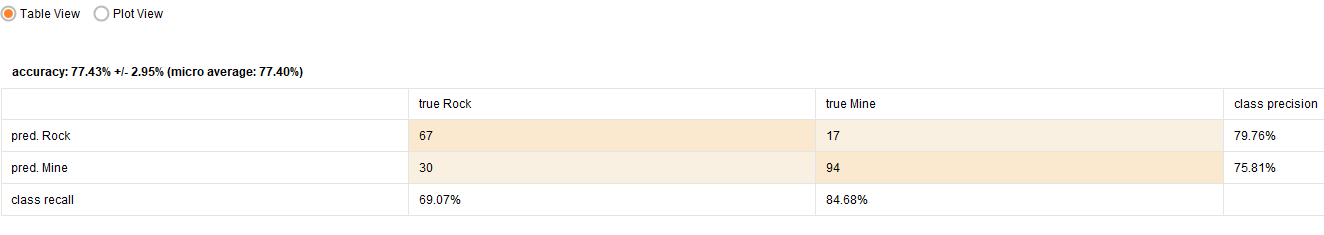
Se obtienen los mismos resultados con o sin la corrección de Laplace.

## 

## 

## Ejercicio 2: Forward selection

Se entrena un modelo con los predictores seleccionados con la heurística forward selection y el algoritmo naive bayes.



Atributos seleccionados: 12, 15, 17, 18

El algoritmo encuentra un máximo local (atributo 12) en 4 iteraciones.

## Ejercicio 3: Backward Elimination

Se entrena un modelo con los predictores seleccionados con la heurística backward elimination y el algoritmo naive bayes.



Atributos descartados: 3, 14, 20, 36, 47, 48, 52, 59

## Ejercicio 4

Se entrena un modelo con los predictores seleccionados con una estrategia evolutiva y el algoritmo naive bayes.





**Análisis personal de los resultados**

La mejor exactitud fue obtenida por el modelo en el que se utilizó el enfoque genético para la selección de atributos. Igualmente, considero que se eligieron demasiados atributos, muchos de los cuales no hacen más que incluir ruido en el modelo.

Debido a esto, decido indagar más en los parámetros del bloque optimize selection y encuentro una opción con la que se puede limitar la cantidad de atributos seleccionados. Escojo el valor 10. Los resultados son los siguientes.

10 atributos:

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media Tabla

Descripción generada automáticamente

Se obtuvieron mejores resultados y los atributos seleccionados tienen más sentido cuando se los compara con los intervalos en los que las 2 gráficas (mine y rock) están más separadas.